WIKIPEDIA

Honeypot (computing)

In computer terminology, a **honeypot** is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honeypot consists of data (for example, in a network site) that appears to be a legitimate part of the site which contains information or resources of value to attackers. It is actually isolated, monitored, and capable of blocking or analyzing the attackers. This is similar to police sting operations, colloquially known as "baiting" a suspect.[1]

The main use for this network decoy is to distract potential attackers from more important information and machines on the real network, learn about the forms of attacks they can suffer, and examine such attacks during and after the exploitation of a honeypot. It provides a way to prevent and see vulnerabilities in a specific network system. A honeypot is a decoy used to protect a network from present or future attacks.[2][3] Honeypots derive their value from the use by attackers. If not interacted with, the honeypot has little to no value. Honeypots can be used for everything from slowing down or stopping automated attacks, capturing new exploits, to gathering intelligence on emerging threats or early warning and prediction.[4]

Types

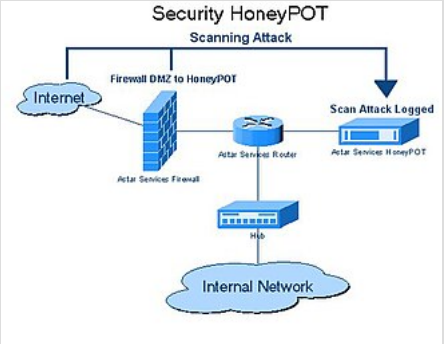

Diagram of an information system honeypot

Honeypots can be differentiated based on if they are physical or virtual:[2][3]

- Physical honeypots: real machine with its own IP address, this machine simulates behaviors modeled by the system. Many times this modality is not used as much as the high price of acquiring new machines, their maintenance and the complication affected by configuring specialized hardware[2][3]
- Virtual honeypots: the use of these types of honeypot allow one to install and simulate hosts on the network from different operating systems, but in order to do so, it is necessary to simulate the TCP/IP of the target operating system. This modality is more frequent.[2][3]

Honeypots can be classified based on their deployment (use/action) and based on their level of involvement. Based on deployment, honeypots may be classified as:[5]

- production honeypots
- research honeypots

Production honeypots are easy to use, capture only limited information, and are used primarily by corporations. Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, which are easier to deploy. They give less information about the attacks or attackers than research honeypots.[5]

Research honeypots are run to gather information about the motives and tactics of the black hat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats that organizations face and to learn how to better protect against those threats.[6] Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations.[7]

Based on design criteria, honeypots can be classified as:[5]

- pure honeypots
- high-interaction honeypots
- low-interaction honeypots

Pure honeypots are full-fledged production systems. The activities of the attacker are monitored by using a bug tap that has been installed on the honeypot's link to the network. No other software needs to be installed. Even though a pure honeypot is useful, stealthiness of the defense mechanisms can be ensured by a more controlled mechanism.

High-interaction honeypots imitate the activities of the production systems that host a variety of services and, therefore, an attacker may be allowed a lot of services to waste their time. By employing virtual machines, multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. In general, high-interaction honeypots provide more security by being difficult to detect, but they are expensive to maintain. If virtual machines are not available, one physical computer must be maintained for each honeypot, which can be exorbitantly expensive. Example: Honeynet.

Low-interaction honeypots simulate only the services frequently requested by attackers.[8] Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the virtual system's security. Example: Honeyd. This type of honeypot was one of the first types being created in the late nineties and was mainly used for detecting attacks, not studying them.[9]

Sugarcane is a type of honeypot that masquerades as an open proxy.[10] It can often take form as a server designed to look like a misconfigured HTTP proxy.[11] Probably the most famous open proxy was the default configuration of sendmail (before version 8.9.0 in 1998) which would forward email to and from any destination.[12]

Deception technology

Recently, a new market segment called deception technology has emerged using basic honeypot technology with the addition of advanced automation for scale. Deception technology addresses the automated deployment of honeypot resources over a large commercial enterprise or government institution.[13]

Malware honeypots

Malware honeypots are a decoy designed to intentionally attract malicious software. It does this by imitating a vulnerable system or network, such as a web server. The honeypot is intentionally set up with security flaws that look to invite these malware attacks. Once attacked IT teams can then analyze the malware to better understand where it comes from and how it acts.[14]

Spam versions

Spammers abuse vulnerable resources such as open mail relays and open proxies. These are servers which accept e-mail from anyone on the Internet—including spammers—and send it to its destination. Some system administrators have created honeypot programs that masquerade as these abusable resources to discover spammer activity.

There are several capabilities such honeypots provide to these administrators, and the existence of such fake abusable systems makes abuse more difficult or risky. Honeypots can be a powerful countermeasure to abuse from those who rely on very high volume abuse (e.g., spammers).

These honeypots can reveal the abuser's IP address and provide bulk spam capture (which enables operators to determine spammers' URLs and response mechanisms). As described by M. Edwards at ITPRo Today:

> Typically, spammers test a mail server for open relaying by simply sending themselves an email message. If the spammer receives the email message, the mail server obviously allows open relaying. Honeypot operators, however, can use the relay test to thwart spammers. The honeypot catches the relay test email message, returns the test email message, and subsequently blocks all other email messages from that spammer. Spammers continue to use the antispam honeypot for spamming, but the spam is never delivered. Meanwhile, the honeypot operator can notify spammers' ISPs and have their Internet accounts canceled. If honeypot operators detect spammers who use open-proxy servers, they can also notify the proxy server operator to lock down the server to prevent further misuse.[15]

The apparent source may be another abused system. Spammers and other abusers may use a chain of such abused systems to make detection of the original starting point of the abuse traffic difficult.

This in itself is indicative of the power of honeypots as anti-spam tools. In the early days of anti-spam honeypots, spammers, with little concern for hiding their location, felt safe testing for vulnerabilities and sending spam directly from their own systems. Honeypots made the abuse riskier and more difficult.

Spam still flows through open relays, but the volume is much smaller than in 2001-02. While most spam originates in the U.S.,[16] spammers hop through open relays across political boundaries to mask their origin. Honeypot operators may use intercepted relay tests to recognize and thwart attempts to relay spam through their honeypots. "Thwart" may mean "accept the relay spam but decline to deliver it." Honeypot operators may discover other details concerning the spam and the spammer by examining the captured spam messages.

Open-relay honeypots include Jackpot, written in Java by Jack Cleaver; *smtpot.py*, written in Python by Karl A. Krueger;[17] and spamhole, written in C.[18] The *Bubblegum Proxypot* is an open-source honeypot (or "proxypot").[19]

Email trap

An email address that is not used for any other purpose than to receive spam can also be considered a spam honeypot. Compared with the term "spamtrap", the term "honeypot" might be more suitable for systems and techniques that are used to detect or counterattack probes. With a spamtrap, spam arrives at its destination "legitimately"—exactly as non-spam email would arrive.

An amalgam of these techniques is Project Honey Pot, a distributed, open source project that uses honeypot pages installed on websites around the world. These honeypot pages disseminate uniquely tagged spamtrap email addresses and spammers can then be tracked—the corresponding spam mail is subsequently sent to these spamtrap e-mail addresses.[20]

Database honeypot

Databases often get attacked by intruders using SQL injection. As such activities are not recognized by basic firewalls, companies often use database firewalls for protection. Some of the available SQL database firewalls provide/support honeypot architectures so that the intruder runs against a trap database while the web application remains functional.[21]

Industrial Control Systems honeypot

Industrial Control Systems (ICS) are often the target of cyberattacks.[22] One of the main targets within ICS are Programmable Logic Controllers.[23] In order to understand intruders' techniques in this context, several

honeypots has been proposed. Conpot [24][25] is a low interaction honeypot capable of simulation Siemens PLCs. HoneyPLC is a medium interaction honeypot that can simulate Siemens, Rockwell and other PLC brands.[26][27]

Honeypot detection

Just as honeypots are weapons against spammers, honeypot detection systems are spammer-employed counter-weapons. As detection systems would likely use unique characteristics of specific honeypots to identify them, such as the property-value pairs of default honeypot configuration,[28] many honeypots in use utilise a set of unique characteristics larger and more daunting to those seeking to detect and thereby identify them. This is an unusual circumstance in software; a situation in which "versionitis" (a large number of versions of the same software, all differing slightly from each other) can be beneficial. There's also an advantage in having some easy-to-detect honeypots deployed. Fred Cohen, the inventor of the Deception Toolkit, argues that every system running his honeypot should have a deception port which adversaries can use to detect the honeypot.[29] Cohen believes that this might deter adversaries. Honeypots also allow for early detection of legitimate threats. No matter how the honeypot detects the exploit, it can alert you immediately to the attempted attack.[30]

Risks

The goal of honeypots is to attract and engage attackers for a sufficiently long period to obtain high-level Indicators of Compromise (IoC) such as attack tools and Tactics, Techniques, and Procedures (TTPs). Thus, a honeypot needs to emulate essential services in the production network and grant the attacker the freedom to perform adversarial activities to increase its attractiveness to the attacker. Although the honeypot is a controlled environment and can be monitored by using tools such as honeywall,[31] attackers may still be able to use some honeypots as pivot nodes to penetrate production systems.[32]

The second risk of honeypots is that they may attract legitimate users due to a lack of communication in large-scale enterprise networks. For example, the security team who applies and monitors the honeypot may not disclose the honeypot location to all users in time due to the lack of communication or the prevention of insider threats.[33][34]

Honey nets

Two or more honeypots on a network form a *honey net*. Typically, a honey net is used for monitoring a larger and/or more diverse network in which one honeypot may not be sufficient. Honey nets and honeypots are usually implemented as parts of larger network intrusion detection systems. A *honey farm* is a centralized collection of honeypots and analysis tools.[35]

The concept of the honey net first began in 1999 when Lance Spitzner, founder of the Honeynet Project, published the paper "To Build a Honeypot".[36]

> "A 'honey net' is a network of high interaction honeypots that simulates a production network and configured such that all activity is monitored, recorded and in a degree, discreetly regulated."

History

An early formulation of the concept, called "entrapment", is defined in FIPS 39 (1976) as "the deliberate planting of apparent flaws in a system for the purpose of detecting attempted penetrations or confusing an intruder about which flaws to exploit".[37]

The earliest honeypot techniques are described in Clifford Stoll's 1989 book *The Cuckoo's Egg*.

> -Lance Spitzner, Honeynet Project

One of the earliest documented cases of the cybersecurity use of a honeypot began in January 1991. On January 7, 1991, while he worked at AT&T Bell Laboratories Cheswick observed a criminal hacker, known as a cracker, attempting to obtain a copy of a password file. Cheswick wrote that he and colleagues constructed a "chroot "Jail" (or "roach motel")" which allowed them to observe their attacker over a period of several months.[38]

In 2017, Dutch police used honeypot techniques to track down users of the darknet market Hansa.

The metaphor of a bear being attracted to and stealing honey is common in many traditions, including Germanic, Celtic, and Slavic. A common Slavic word for the bear is *medved* "honey eater". The tradition of bears stealing honey has been passed down through stories and folklore, especially the well known Winnie the Pooh.[39][40]

See also

- Canary trap
- Client honeypot
- Cowrie
- Defense strategy (computing)
- HoneyMonkey
- Honeytoken
- Network telescope
- Operation Trust
- Tarpit

References and notes

1. Cole, Eric; Northcutt, Stephen. "Honeypots: A Security Manager's Guide to Honeypots" (https://web.archive.org/web/20170316110416/https://www.sans.edu/cyber-research/security-laboratory/article/honeypots-guide). Archived from the original (https://www.sans.edu/cyber-research/security-laboratory/article/honeypots-guide) on 16 March 2017.
2. Provos, N. "A Virtual Honeypot Framework" (https://www.usenix.org/legacy/event/sec04/tech/full_papers/provos/provos_html/). *USENIX*. Retrieved 29 April 2023.
3. Mairh, A; Barik, D; Verma, K; Jena, D (2011). "Honeypot in network security: A survey". *Proceedings of the 2011 International Conference on Communication, Computing & Security - ICCCS '11* (https://dl.acm.org/doi/abs/10.1145/1947940.1948065). Vol. 1. pp. 600–605. doi:10.1145/1947940.1948065 (https://doi.org/10.1145%2F1947940.1948065). ISBN 978-1-4503-0464-1. S2CID 12724269 (https://api.semanticscholar.org/CorpusID:12724269). Retrieved 29 April 2023.
4. Spitzner, L. (2003). "Honeypots: Catching the insider threat" (https://dx.doi.org/10.1109/csac.2003.1254322). *19th Annual Computer Security Applications Conference, 2003. Proceedings*. IEEE. pp. 170–179. doi:10.1109/csac.2003.1254322 (https://doi.org/10.1109%2Fcsac.2003.1254322). ISBN 0-7695-2041-3. S2CID 15759542 (https://api.semanticscholar.org/CorpusID:15759542).

5. Mokube, Iyatiti; Adams, Michele (March 2007). "Honeypots: Concepts, approaches, and challenges" (https://doi.org/10.1145/1233341.1233399). *Proceedings of the 45th annual southeast regional conference*. pp. 321–326. doi:10.1145/1233341.1233399 (https://doi.org/10.1145%2F1233341.1233399). ISBN 9781595936295. S2CID 15382890 (https://api.semanticscholar.org/CorpusID:15382890).
6. Lance Spitzner (2002). *Honeypots tracking hackers*. Addison-Wesley. pp. 68–70. ISBN 0-321-10895-7.
7. Katakoglu, Onur (2017-04-03). "Attacks Landscape in the Dark Side of the Web" (http://www.madlab.it/papers/sac17_darknets.pdf) (PDF). *acm.org*. Retrieved 2017-08-09.
8. Litchfield, Samuel; Formby, David; Rogers, Jonathan; Meliopoulos, Sakis; Beyah, Raheem (2016). "Rethinking the Honeypot for Cyber-Physical Systems" (https://ieeexplore.ieee.org/document/7676152). *IEEE Internet Computing*. **20** (5): 9–17. doi:10.1109/MIC.2016.103 (https://doi.org/10.1109%2FMIC.2016.103). ISSN 1089-7801 (https://search.worldcat.org/issn/1089-7801). S2CID 1271662 (https://api.semanticscholar.org/CorpusID:1271662).
9. Göbel, Jan Gerrit; Dewald, Andreas; Freiling, Felix (2011). *Client-Honeypots* (https://dx.doi.org/10.1524/9783486711516). doi:10.1524/9783486711516 (https://doi.org/10.1524%2F9783486711516). ISBN 978-3-486-71151-6.
10. Talukder, Asoke K.; Chaitanya, Manish (17 December 2008). *Architecting Secure Software Systems Page 25 – CRC Press, Taylor & Francis Group* (https://books.google.com/books?id=ntsJqzfwFhkC&dq=honeypot+sugarcane&pg=PA25). CRC Press. ISBN 9781420087857.
11. "Exposing the Underground: Adventures of an Open Proxy Server" (https://www.secureworks.com/blog/proxies). 21 March 2011.
12. "Capturing web attacks with open proxy honeypots" (https://lwn.net/Articles/240120/). 3 July 2007.
13. "Deception related technology – its not just a "nice to have", its a new strategy of defense – Lawrence Pingree" (http://blogs.gartner.com/lawrence-pingree/2016/09/28/deception-related-technology-its-not-just-a-nice-to-have-its-a-new-strategy-of-defense/). 28 September 2016.
14. Praveen (2023-07-31). "What Is a Honeypot in Cybersecurity? Types, Implementation, and Real-World Applications" (https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/what-are-honeypots-benefits-types/). *Cybersecurity Exchange*. Retrieved 2023-12-05.
15. Edwards, M. "Antispam Honeypots Give Spammers Headaches" (https://web.archive.org/web/20170701040344/http://windowsitpro.com/exchange-server/antispam-honeypots-give-spammers-headaches). Windows IT Pro. Archived from the original (http://windowsitpro.com/exchange-server/antispam-honeypots-give-spammers-headaches) on 1 July 2017. Retrieved 11 March 2015.
16. "Sophos reveals latest spam relaying countries" (http://www.net-security.org/secworld.php?id=4085). *Help Net Security*. 24 July 2006. Retrieved 14 June 2013.
17. "Honeypot Software, Honeypot Products, Deception Software" (https://web.archive.org/web/20031008120110/http://www.honeypots.net/honeypots/products). *Intrusion Detection, Honeypots and Incident Handling Resources*. Honeypots.net. 2013. Archived from the original (http://www.honeypots.net/honeypots/products) on 8 October 2003. Retrieved 14 June 2013.
18. dustintrammell (27 February 2013). "spamhole – The Fake Open SMTP Relay Beta" (http://sourceforge.net/projects/spamhole/). *SourceForge*. Dice Holdings, Inc. Retrieved 14 June 2013.
19. Ec-Council (5 July 2009). *Certified Ethical Hacker: Securing Network Infrastructure in Certified Ethical Hacking* (https://books.google.com/books?id=nERI0SQqF_sC&pg=SA3-PA23). Cengage Learning. pp. 3–. ISBN 978-1-4354-8365-1. Retrieved 14 June 2013.
20. "What is a honeypot?" (https://www.ionos.com/digitalguide/server/security/honeypots-it-security-through-decoy-programs/). *IONOS Digital Guide*. 8 August 2017. Retrieved 2022-10-14.

21. "Secure Your Database Using Honeypot Architecture" (https://web.archive.org/web/20120308171843/http://www.dbcoretech.com/?p=453). dbcoretech.com. August 13, 2010. Archived from the original (http://www.dbcoretech.com/?p=453) on March 8, 2012.
22. Langner, Ralph (May 2011). "Stuxnet: Dissecting a Cyberwarfare Weapon" (https://ieeexplore.ieee.org/document/5772960). *IEEE Security & Privacy*. **9** (3): 49–51. doi:10.1109/MSP.2011.67 (https://doi.org/10.1109%2FMSP.2011.67). ISSN 1558-4046 (https://search.worldcat.org/issn/1558-4046). S2CID 206485737 (https://api.semanticscholar.org/CorpusID:206485737).
23. Stouffer, Keith; Falco, Joe; Scarfone, Karen (June 2011). "Guide to Industrial Control Systems (ICS) Security - Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC)" (https://dx.doi.org/10.6028/nist.sp.800-82). *NIST Publications* (NIST Special Publication (SP) 800-82). Gaithersburg, MD: 155 pages. doi:10.6028/nist.sp.800-82 (https://doi.org/10.6028%2Fnist.sp.800-82).
24. Jicha, Arthur; Patton, Mark; Chen, Hsinchun (September 2016). "SCADA honeypots: An in-depth analysis of Conpot" (https://ieeexplore.ieee.org/document/7745468). *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. pp. 196–198. doi:10.1109/ISI.2016.7745468 (https://doi.org/10.1109%2FISI.2016.7745468). ISBN 978-1-5090-3865-7. S2CID 14996905 (https://api.semanticscholar.org/CorpusID:14996905).
25. *Conpot* (https://github.com/mushorg/conpot), MushMush, 2023-06-23, retrieved 2023-06-24
26. López-Morales, Efrén; Rubio-Medrano, Carlos; Doupé, Adam; Shoshitaishvili, Yan; Wang, Ruoyu; Bao, Tiffany; Ahn, Gail-Joon (2020-11-02). "HoneyPLC: A Next-Generation Honeypot for Industrial Control Systems" (https://dl.acm.org/doi/10.1145/3372297.3423356). *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. CCS '20. New York, NY, USA: Association for Computing Machinery. pp. 279–291. doi:10.1145/3372297.3423356 (https://doi.org/10.1145%2F3372297.3423356). hdl:2286/R.I.57069 (https://hdl.handle.net/2286%2FR.I.57069). ISBN 978-1-4503-7089-9. S2CID 226228191 (https://api.semanticscholar.org/CorpusID:226228191).
27. *HoneyPLC* (https://github.com/sefcom/honeyplc), SEFCOM, 2023-05-24, retrieved 2023-06-24
28. Cabral, Warren; Valli, Craig; Sikos, Leslie; Wakeling, Samuel (2019). "Review and Analysis of Cowrie Artefacts and Their Potential to be Used Deceptively". *Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence*. IEEE. pp. 166–171. doi:10.1109/CSCI49370.2019.00035 (https://doi.org/10.1109%2FCSCI49370.2019.00035). ISBN 978-1-7281-5584-5.
29. "Deception Toolkit" (http://all.net/dtk/index.html). *All.net*. 2013. Retrieved 14 June 2013.
30. *Honeypots for Windows* (https://dx.doi.org/10.1007/978-1-4302-0007-9). 2005. doi:10.1007/978-1-4302-0007-9 (https://doi.org/10.1007%2F978-1-4302-0007-9). ISBN 978-1-59059-335-6.
31. "Honeywall CDROM – The Honeynet Project" (https://web.archive.org/web/20221011002345/https://www.honeynet.org/projects/old/honeywall-cdrom/). Archived from the original (https://www.honeynet.org/projects/old/honeywall-cdrom/) on 2022-10-11. Retrieved 2020-08-07.
32. Spitzner, Lance (2002). *Honeypots Tracking Hackers*. Addison-Wesley Professional. OCLC 1153022947 (https://search.worldcat.org/oclc/1153022947).
33. Qassrawi, Mahmoud T.; Hongli Zhang (May 2010). "Client honeypots: Approaches and challenges" (https://ieeexplore.ieee.org/document/5488508). *4th International Conference on New Trends in Information Science and Service Science*: 19–25.
34. "illusive networks: Why Honeypots are Stuck in the Past | NEA | New Enterprise Associates" (https://www.nea.com/blog/illusive-networks-why-honeypots-are-stuck-in-the-past). *www.nea.com*. Retrieved 2020-08-07.

35. "cisco router Customer support" (https://web.archive.org/web/20170116043827/http://www.reouterhelpsupport.com/cisco-customer-service.php). Clarkconnect.com. Archived from the original (http://www.reouterhelpsupport.com/cisco-customer-service.php) on 2017-01-16. Retrieved 2015-07-31.
36. "Know Your Enemy: GenII Honey Nets Easier to deploy, harder to detect, safer to maintain" (https://web.archive.org/web/20090125224729/http://old.honeynet.org/papers/gen2/). *Honeynet Project*. 12 May 2005. Archived from the original (http://old.honeynet.org/papers/gen2/) on 25 January 2009. Retrieved 14 June 2013.
37. "National Bureau of Standards (February 15, 1976). Glossary for Computer Systems Security" (https://www.govinfo.gov/content/pkg/GOVPUB-C13-18320c963d272d740d6dffce808fce3d/pdf/GOVPUB-C13-18320c963d272d740d6dffce808fce3d.pdf) (PDF). *www.govinfo.gov*. Retrieved 19 Mar 2023.
38. "An Evening with BerferdIn Which a Cracker is Lured, Endured, and Studied" (http://cheswick.com/ches/papers/berferd.pdf) (PDF). *cheswick.com*. Retrieved 3 Feb 2021.
39. "The word for "bear" " (https://web.archive.org/web/20130929171327/http://www.pitt.edu/~votruba/qsonhist/bearetymologyslovakenglishwelsh.html). *Pitt.edu*. Archived from the original (http://www.pitt.edu/~votruba/qsonhist/bearetymologyslovakenglishwelsh.html) on 29 September 2013. Retrieved 12 Sep 2014.
40. Shepard, E. H., Milne, A. A. (1994). The Complete Tales of Winnie-the-Pooh. United Kingdom: Dutton Children's Books.

Further reading

- Lance Spitzner (2002). *Honeypots tracking hackers*. Addison-Wesley. ISBN 0-321-10895-7.
- Sean Bodmer; Max Kilger; Gregory Carpenter; Jade Jones (2012). *Reverse Deception: Organized Cyber Threat Counter-Exploitation*. McGraw-Hill Education. ISBN 978-0-07-177249-5.

External links

- The Ultimate Fake Access Point (https://rootsh3ll.com/ultimate-fake-access-point-walkthrough/) Archived (https://web.archive.org/web/20210225223201/https://rootsh3ll.com/ultimate-fake-access-point-walkthrough/) 2021-02-25 at the Wayback Machine - AP less clear-text WPA2 passphrase hacking
- Distributed Open Proxy Honeypots Project: WASC (http://projects.webappsec.org/w/page/29606603/Distributed%20Web%20Honeypots)
- SANS Institute: What is a Honey Pot? (https://web.archive.org/web/20090918210959/http://www.sans.org/resources/idfaq/honeypot3.php)
- SANS Institute: Fundamental Honeypotting (http://www.sans.org/reading_room/whitepapers/detection/fundamental-honeypotting_2054)
- Project Honeypot (http://www.projecthoneypot.org/)
- A curated list of honeypots, tools and components focused on open source projects (https://github.com/paralax/awesome-honeypots#honeypots/)